Towards Believable Behavior Generation for Embodied Conversational Agents

Andrea Corradini[1], Morgan Fredriksson[2], Manish Mehta[1], Jurgen Königsmann[2], Niels Ole Bernsen[1], and Lasse Johannesson[2]

[1] Natural Interactive Systems Laboratory
University of Southern Denmark, 5230 Odense M, Denmark
{andrea,manish,nob}@nis.sdu.dk
http://www.nis.sdu.dk/staff
[2] Liquid Media AB
Skaanegatan 101, 11635 Stockholm, Sweden
{morgan,jurgen,lasse}@liquid.se
http://www.liquid.se

Abstract. This paper reports on the generation of coordinated multimodal output for the NICE (Natural Interactive Communication for Edutainment) system [1]. In its first prototype, the system allows for fun and experientially rich interaction between primarily 10 to 18 years old human users and 3D-embodied fairy tale author H.C. Andersen in his study. User input consists of domain-oriented spoken conversation combined with 2D input gesture, entered via a mouse-compatible device. The animated character can move about and interact with his environment as well as communicate with the user through spoken conversation and non-verbal gesture, body posture, facial expression and gaze. The described approach aims to make the virtual agent's appearance, voice, actions, and communicative behavior convey the impression of a character with human-like behavior, emotions, relevant domain knowledge, and a distinct personality. We propose an approach to multimodal output generation, which exploits a richly parameterized semantic instruction from the conversation manager and splits the instruction into synchronized text instructions to the text-to-speech synthesizer, and behavioral instructions to the animated character. Based on the implemented version of this approach, we are in the process of creating a behavior sub-system that combines the described multimodal output instructions with parameters representing the current emotional state of the character, producing animations that express emotional state through speech and non-verbal behavior.

1 Introduction

CASA (Computers Are Social Actors) studies [2] have shown that humans automatically and unconsciously tend to treat computers and other new media as real social entities. When interacting with computer systems that are able to display attitude and personality, people find them polite, extrovert, etc. Humans respond to these affective stimuli and traits in the same way as they do to people in relation to the politeness, extroversion, or other psycho-social phenomena they have perceived. These findings

M. Bubak et al. (Eds.): ICCS 2004, LNCS 3038, pp. 946–953, 2004.

advocate the Media Equation: media = real life, which essentially postulates that interaction between humans and machines tends to be social in nature, i.e. can be viewed as a specialization of the anthropomorphic tendency of human beings. In this light, it has been argued that Human-Computer Interaction (HCI) is fundamentally social and thus that social rules governing human-human communication apply to HCI as well. Consequently, the user interface of interactive technologies can be improved by leveraging human expectations of natural human social features.

Supported by these trends, and by today's advances in human language technologies and increasingly powerful computer graphics techniques that are making available a broader set of tools to create more flexible, human-centered and adaptive user interfaces, there has been a thrust towards designing agent based interfaces which exhibit human-like behavior and appearance. These interfaces, termed embodied conversational agents (ECAs) [3], aim both to use and to realize cues inherently peculiar to human-human communication, such as sense of presence, mixed initiative and non-verbal behaviors to hold up their end of the dialogue with the user.

We have implemented a domain-oriented (non-task-oriented [4]) conversation system that allows users to interact in a natural, fun, and experientially rich manner with an embodied conversational character impersonating the Danish fairytale writer Hans Christian Andersen (HCA) within a graphical 3D environment. The system accepts spoken and 2D gestural input entered via a mouse-compatible device, recognizes, interprets, and fuses these modalities with context information, and eventually generates an appropriate coherent behavior in response to the user input.

In this paper, we report on our approach to generating coordinated multimodal output with the aim of making the virtual agent's appearance, voice, actions, and communicative behavior believable, i.e. conveying the impression of a character with human-like behavior, emotions, relevant domain knowledge, and a distinct personality, entertaining, and instructive, i.e. providing true historical information, to the user.

2 Multimodal Output Generation

2.1 Related Work

Other researchers have developed embodied multimodal agents that produce natural conversation in task-oriented applications [3].

Rea [18] plays the role of a real estate salesperson; she uses gaze, head movements and facial expressions for functions such as turn-taking, emphasis and greetings. The verbal and non-verbal communication aspect of Rea is limited to task oriented dialogue in which the agent interacts with users to determine their needs, shows them around virtual properties, and attempts to sell them a house. Baldi [19] is an animated conversational agent, who helps students accurately produce expressive speech. The interactive system's curriculum development software lets teachers and students customize class work. The animated agent communication is limited to facial movements that are synchronized to its audible speech. Greta [20] responds to user queries by speaking and exhibiting gaze, head movements, and facial expression. The conversation simulated is restricted to a limited task.

Also in the game industry there have been attempts to provide the user with different modalities in order to provide a rich interaction experience. Black and White [21]

is a strategy game developed by Lionhead Studios Ltd., which involves drawing symbols on the screen to cast miracles such as a rain spell to increase food production. The shape of the symbols ranges from arrows and swirls to letters and numbers. As the user advances in the game, the symbols she has to draw become more complex, making it harder to cast them quickly during gameplay. Arx Fatalis [22] is a first-person role-playing game that allows the player to draw burning runes in mid-air using the mouse. A series of these gestures combine to create powerful magic spells that will protect the player or empower him to defeat his enemies and pursue his quest. These games provide unimodal user interaction yet there have been rather few attempts to enhance user experience with multimodal input modalities. The same goes for synchronization of verbal and non-verbal output from the player and non-player character.

2.2 Response Generator

The HCA character module is always in one of three output states. While in the Communicative Function (CF) output state, the character shows in his behavior that he is aware of being spoken to or addressed by the user. A Non-Communicative Action (NCA) output state refers to the situation in which HCA is not engaged in conversation with a user. The agent is in a Communicative Action (CA) output state anytime he takes the turn in producing outputs that serve as a vehicle for the on-going conversation. CA output is the character's actual conversational contribution in the form of verbal realization of one or more speech acts within a single turn, physical actions within the graphical environment, emotion display, gaze and gestural behaviors. Responses to, and questions for, the user, observations, confirmations and acknowledgements, and meta-communication, such as clarification questions, are produced by HCA when in this state.

Similarly to non-verbal output categorization in [5], we have employed a common strategy to produce output behaviors for both NCA and CF states while we have developed a response generator module to explicitly treat CA state output. For NCA and CF states, we have defined approximately 30 and 15 behaviors, respectively. CF states are generated from a pre-defined subset of elementary behaviors. Finite-state machines whose states are also elementary behaviors are instead utilized to generate NCA states due to the indefinitely long sequence of actions required while in such state. Most of these behaviors are non-verbal, yet a few have also a non-speech audio component, e.g., for playing footsteps sound when the character is moving about in his study. NCA behaviors are typically state-of-the-mind expressions, such as being idle or thinking, and physical actions, such as picking up an object or dropping one. CF behaviors are feedback gestures, e.g., nodding, and posture changes, e.g., clasping hands. Since the CF and NCA output repertoire is fixed at design-time, its realization does not need to involve the response generator. CF and NCA behaviors are randomly selected at certain time intervals and sent directly to the graphic engine for realization.

As concerns CA output, giving HCA a richer persona through emotion and personality modeling is not sufficient to creating a life-like character [6,7]. To increase the character's believability [8], it also has to be able to display human-like behavior by combining non-verbal (gesture, facial expression, pose and gaze) and verbal (speech) output in a consistent way [9]. Coherence is a general rule in social human-human communication [10]. Hence, following the Media Equation, we can expect this rule to

Fig. 1. *(left to right)* HCA performing a DONT_KNOW animation while in the CA output state

apply to human-machine interfaces as well. Furthermore, as people dislike interacting with individuals who behave incoherently [11,2], consistency may also contribute to increasing user effectiveness and satisfaction.

Our response generator links emotional patterns and character animation in terms of speech and non-verbal communication. It performs in real-time and enables generation of a comprehensive representation of communicative actions which can be rendered via speech synthesis and by the animation component (see Fig. 1).

In its technical implementation, the response generator receives from the character module a parameterized semantic instruction composed of input values, text-to-speech (TTS) references, and/or references to non-verbal behaviors. The TTS references are used to retrieve text template output with embedded start and end tags for non-verbal behaviors (bookmarks) that are stored in a form such as: "*I [g0] don't remember exactly [/g0] when I wrote {FAIRYTALE}"*. In this example, elements within square brackets starting with numbered 'g' letters represent onset and offset of non-specified non-verbal parts of the template. Elements within curly parentheses, like *FAIRYTALE,* are placeholders for variable values to be filled in using input value information delivered by the natural language understanding module [12] during a conversational turn. 'The Princess and the Pea' and 'The Little Mermaid' are two possible values in the present example (see [12] for details on the structure of conversation in NICE). Both verbal variable values, as was just mentioned, and non-verbal behavioral elements are initially uninstantiated in the sense that they need to be retrieved at run-time. This approach allows for a high degree of flexibility as the binding of non-verbal behavior to speech occurs at run-time rather than being hard-coded, enabling a sentence to be synthesized at different times with, e.g., different accompanying gestures.

In order to provide timing information for speech and gesture during rendering, non-verbal behavioral elements are made up of two sets of tags to indicate their start and end, respectively. Thus, in the example above, tags *[g0]* and *[/g0]* indicate that a single movement has to co-occur with uttering the spoken text '*don't remember exactly'* around which they are wrapped. Any non-verbal behavior can be attached to the gestural behavior *[g0]* while uttering the short text.

Once non-verbal-behavior tags have been processed and variable values inserted into the templates, one surface language string results. This string is sent to the speech synthesizer, which synthesizes the verbal output and, whenever it meets a bookmark, sends a message to the response generator. The response generator creates an XML representation of the non-verbal element and sends it to the animation engine that takes care of the graphics output. Fig. 1 shows a series of snapshots of the animation generated with the text template used as example, while the XML representation of the non-verbal element in this case simply looks like:

```
<play>
    <animation>
            <name>DONT_KNOW</name>
            <startTime>0</startTime>
            <stopTime></stopTime>
            <data>H, 0; E, 200; J, 500; Sil, 800</data>
    </animation>
</play>
```

The item `<name>` indicates the name of the animation to play, among those that are loaded in memory upon start-up by the rendering application. We refer to these animations as elementary or primitive animations. The `<data>` tag is used to specify additional animation-related information while `<startTime>` provides timing information for the start of the animation. The optional item `<stopTime>` is not being used because we did not want to explicitly set the duration of the animation.

In addition to elementary animations, more complex non-verbal behaviors can be created, assigned a name, and stored by the response generator. To create a new animation, its behavior must be defined in terms of existing animations. For example, assuming the existence of an animation called PUFF, a new composite animation GIVE_UP_FRUSTRATED can be created using PUFF and DONT_KNOW. Because the rendering engine recognizes only elementary animations, to play the new animation, the response generator has to create as many different XML representation strings as the number of primitive animations used in the description of the new one, i.e. two for GIVE_UP_FRUSTRATED. Sequentiality, parallelism and partial overlapping of existing animations to create the new animation can be tuned by setting appropriate values for the temporal items in the XML representation. So, GIVE_UP_FRUSTRATED can occur either as sequence of DONT_KNOW and PUFF, or vice versa, or by having these two running simultaneously, etc.

In the our first prototype, we use approximately 300 spoken utterance templates, many of which are no-variable stories to be told by HCA, and 100 different non-verbal behavior primitives that we identified after analyzing a 4-hour video recording of a human actor performing as HCA in a children theater in the city of Odense, Denmark.

2.3 Graphical Engine and Animations

The rendering module consists of a number of different subsystems that take care of different aspects of the animation process. These are the following.

Visual System. Handles everything related to on-screen visualization, such as rendering of the 3D-environment and the characters as well as output of 2D-text.

XML-based Entity Factory. A set of XML files loaded at start-up to specify which components, such as characters, animations and objects, are included and how they are configured.

Input Handler and Object Tracker. Handle the user's keyboard and mouse input. For each input device position, the object tracker keeps track of the objects in the scenery. This is important as some objects in the scene are 'active', i.e. can be manipulated, e.g. picked up, dropped, or pointed at, by the character.

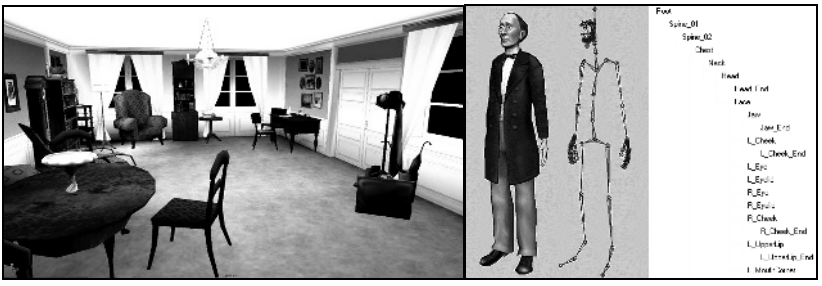

Fig. 2. HCA's study: *(left)* general view, *(right)* skinned mesh of the HCA character

Network Handler. For TCP-IP or UDP socket communication with other modules.

Navigation System. Handles the movement of characters, stops them from walking through walls or colliding with objects in the environment.

3D Sound System. Handles playback of environmental sounds as well as character speech; the sound can be global or positioned.

Scheduler. Manages events like synchronization of animations, collisions checking, rendering etc. For example, it is used to synchronize the start of animations at a specific time and checking for collisions every Nth frame while rendering is done at every frame. Events in the scheduler are either framebased or time-based.

Animation and Camera System. Handle the updating of characters (currently we only have the HCA character, yet more will be added soon) and other animated objects and keep track of all cameras (currently there are five views) used by the application. The camera decides which part of the 3D scene is rendered. To animate a three-dimensional character, we change its position, scale and orientation at different points in time. The animation system uses three scalar values to represent position and scale, and quaternion values to represent orientation. A character is built upon a hierarchy of elementary parts, referred to as frames, where each single frame represents a bone in the character. The hierarchy of frames, together with a textured polygon mesh and skin weighting information, is represented as a skinned mesh (Fig. 2). The skinning information specifies the influence a frame has on its mesh. To avoid breaking up the mesh while animating the characters, we use vertex blending. The root frame contains a transformation matrix relative to the world space. An animation that affects the root node affects the whole scene while one that affects a leaf node does not affect any other node. Hence, the frame hierarchy gives the system the ability of playing single animations in parallel for different parts of the body to obtain complex animations.

Each animation is given a priority index. For example, let us assume that we set the priorities of animation IDLE to 0, WALK to 5 and NOD to 50. The WALK animation, which affects all nodes in the frame hierarchy, would replace the IDLE animation completely while NOD would only affect the nodes from the neck and down (Fig. 2). NOD with its higher priority overloads only the relevant nodes while WALK or IDLE affects the rest of the hierarchy. The WALK animation overloads all nodes of the IDLE animation. The result is a walking man nodding his head.

The animation system works as a sequencer, it receives network commands and schedules animation events via the scheduler system. The animations can be started in parallel, having, e.g., the same start time. The graphical engine uses its own methods for memory allocation to facilitate the tracing of memory usage and leaks.

3 Conclusions and Future Work

Most of the information exchanged in human face-to-face communication takes place over the non-verbal channel. Therefore, when it comes to developing embodied conversational agents, non-verbal representation is at least as important as verbal representation. Unfortunately, defining an exhaustive representation that encapsulates the attributes necessary for ECA behavior remains an unsolved problem.

In this paper, we have proposed how to build multimodal output generation for an ECA system that supports agent's believability in displaying full-body human-like behavior. Our system attains its educational goal by providing correct factual information, both visually, via a variety of non-verbal behaviors, notably gesture, body-posture, gaze and emotion, and orally via spoken utterances. Any inconsistency between verbal and non-verbal components may deceive and mislead the user [13]. Thus, coordination between output modalities is a very important step toward this goal. Incoherent output realization is counterproductive as it undermines the user's learning process rather than reinforcing it. The system attains its entertainment goal as well. Our character is lifelike, reproduces the human physics in detail, and performs non-verbal behaviors in an exaggerated manner as this has been proven to convey emotions more efficiently and directly than regular performance (in fact, caricaturists take advantage of this), making interaction a fun experience. Embodiment enhances entertainment and effective user engagement [14]. A preliminary system test, we recently run with eighteen 10 to 18 years old kids, seems to support these facts.

Several XML-based languages have been proposed to specify human communicative behavior [15] but none of these fully capture the non-verbal information conveyed by humans. In our approach, we use a high-level XML formalism to describe the overt form of non-verbal communication by utilizing a parameterized library of a few hand-crafted behaviors that can be flexibly and easily expanded to include new elements for modeling additional behaviors, emotions and moods. The main limitation of our approach is the inherent impossibility to fine-control the motions of single body parts. However, with the capability to combine a set of few elementary behaviors we can cover a large variety of movements, thus reaching a compromise between the number of pre-defined behaviors and the complexity to generate others from them.

The system is undergoing continuous improvement. We plan to deal with animation co-articulation to allow smooth rendering of an animation that is started when another one is not yet concluded. We also plan to focus on the auditory realization of communicative intention. Providing HCA's spoken utterances with stress and intonation will be a first step. Linking facial expression and prosody to further personalize the character's communication of emphasis and topic will be the next step. Moreover, in the effort to improve character believability and, potentially, the perceived quality of spoken segments [16,17], we plan to synchronize speech and lip movements using phoneme-to-viseme mapping.

Acknowledgement. We gratefully acknowledge the support from the EU Human Language Technologies programme under Contract no. IST-2001-35293.

References

1. http://www.niceproject.com
2. Reeves, B., and Nass, C.: The Media Equation: how people treat computers, televisions and new media like real people and places. Cambridge, Cambridge University Press, 1996
3. Cassell, J., Sullivan, J., Prevost, S., and Churchill, E. (eds.): Embodied conversational agents. Cambridge, MA: MIT Press, 2000
4. Bernsen, N.O., Dybkjær, H., and Dybkjær, L.: Designing Interactive Speech Systems. From First Ideas to User Testing. London:Springer Verlag, 1998
5. Beskow J., Edlund, J., and Nordstrand, M.: A model for generalised multi-modal conversation system output applied to an animated talking head. In: Minker, W., et al. (eds.): Spoken Multimodal Human-Computer Conversation in Mobile Envs, Kluwer Academic, 2004
6. Argyle, M.: Bodily Communication. 2nd edition, London and NYC: Methuen & Co., 1986
7. Knapp, M.L.: Non-verbal Communication in Human Interaction. 2nd edition, Holt, Rinehart and Winston Inc., New York City, 1978
8. Loyall, A.B.: Believable Agents: Building Interactive Personalities. PhD thesis, Tech Report CMU-CS-97-123, Carnegie Mellon University, 1997
9. Picard, R.: Affective Computing. MIT Press, 1997
10. Fiske, S.T., and Taylor, S.E.: Social Cognition. New York, McGraw Hill, 1991
11. Nass, C., Isbister, K., and Lee, E.-J.: Truth is beauty: Researching embodied conversational agents. In Cassell, J., et al (eds.):Embodied conversational agents. MIT Press, 374-402, 2000
12. Bernsen, N.O., Charfuelàn, M., Corradini, A., et al.: First Prototype of Conversational H.C. Andersen. In: Proc. of ACM Int'l Working Conf. on Advanced Visual Interfaces, 2004
13. Ekman P., and Friesen, W.V.: Nonverbal leakage and clues to deception. Psychiatry 32, 1969, 88-95
14. Koda, T., and Maes, P.: Agents with faces: The effects of personification of agents. Proceedings of Human-Computer Interaction, London, UK, 1996, 239-245
15. http://www.vhml.org/workshops/AAMAS/papers.html
16. Massaro, D.W., and Cohen, M.: Speech perception in perceivers with hearing loss: Synergy of multiple modalities. Jou. of Speech, Language, and Hearing Res., 42, 1999, 21-41
17. McGurk, H., and MacDonald, J.: Hearing lips and seeing voices. Nature 264, 1976, 746-748
18. Casell, J., Bickmore, J., Billinghurst, M., Campbell L., Chang K., Vilhjalmsson H., and Yan H.: Embodiment in conversational interfaces: Rea In: Proc. of CHI 99, 1999, 520-527
19. Massaro, D.W., Bosseler, A., and Light, J.: Development and Evaluation of a Computer-Animated Tutor for Language and Vocabulary Learning. 15th Int'l Congress of Phonetic Sciences, Barcelona, Spain, 2003
20. Pelachaud, C., Carofiglio, V., De Carolis, B., de Rosis, F., and Poggi, I.: Embodied Contextual Agent in Information Delivering Application, First International Joint Conference on Autonomous Agents & Multi-Agent Systems, Bologna, Italy, 2002
21. http://www.blackandwhite.ea.com/
22. http://www.arxfatalis-online.com/